



Escuela de Ingeniería en Computación
IC 5701 - Compiladores e Intérpretes
Grupo 20

Profesor: Aurelio Sanabria

Proyecto 2
Explorador para el lenguaje "Pandémico"

Jeremy José Live González

2015068507

Juan Pablo Martínez Brenes

2016140404

Luis Ortiz Rua

2016117738

Alonso Obando Carmona

2014006700

I Semestre 2021

Tabla de Contenido

Tabla de Contenido	2
Demostración y discusión	3
Lecciones Aprendidas	4
Creación de componentes léxicos	5
Manejo de errores	8
Completitud del proyecto	9
Salud mental / Amor propio / Diversión / Crecimiento personal	10

Demostración y discusión

En esta sección se demuestra una discusión sobre cómo se gestionó la creación del explorador. De esta forma se muestra por escrito cómo se realizó esta entrega, enseguida se va hablar sobre la creación de este mismo.

Para el desarrollo de este proyecto, se requirió teoría básica sobre compiladores para poder llevarlo a cabo, esto implicó estudiar los documentos y videos suministrados por el profesor, además de analizar el código del explorador del lenguaje Ciruelas para entender mejor la estructura de los datos que se recomienda utilizar para desarrollar un programa de este tipo.

Una vez se esclareció el panorama sobre cómo programar un explorador, que incluye manejo de errores y la programación de la gramática realizada en el entregable anterior, se procedió discutir en grupo la solución, en este punto se acordó utilizar el paradigma orientado a objetos en el lenguaje de programación Python y se estandarizó la forma en que se iba a documentar el código y nombrar las variables.

Durante el desarrollo del programa, se hicieron algunos cambios en la gramática antes definida debido a que como se tenía definido no favorecía al evaluador de tokens, además, se ideó un algoritmo para el manejo de errores, el cual presentó un reto que involucra identificar el error en específico con el que se estaba tratando. Para esto, se escribió una serie de ejemplos en el lenguaje Pandémico, para posteriormente introducir errores manualmente e irlos identificando dentro del explorador.

Lecciones Aprendidas

En esta sección se muestra las opiniones de los integrantes del grupo dando a exponer su pensar junto con su experiencia vivida de lo que fue para cada uno de ellos las lecciones aprendidas en este proyecto realizando el explorador.

Alonso: En el proceso de programación se logró identificar los componentes claves para tomar la gramática del lenguaje y hacer una revisión de un documento creado por un usuario con nuestro lenguaje creado y, si fuera el caso, desplegar errores en torno a esto. Se comprendió la forma en la que se puede realizar un proceso de estos mediante el uso de una serie de funciones implementadas en el lenguaje de Python para simular el análisis real de un compilador.

Juan Pablo: Basados en la implementación de los elementos del BNF del lenguaje Pandémico se logra sustraer y colocar en categorías los componentes léxicos para ser comparados con la escritura del usuario y próximamente validarlas con nuestra sintaxis. Basándonos en el BNF determinamos qué elementos eran importantes y la forma en la que se debían de implementar para poder tener un correcto uso del lenguaje.

Jeremy: Dentro del aprendizaje pudimos determinar los diferentes cambios necesarios de los primeros planteamientos del lenguaje en torno a la implementación, ya que partimos desde el concepto original del lenguaje, pero al momento de implementarlo realizamos varias modificaciones en función de mejorar la interacción con el lenguaje para el usuario y facilitar su lectura por lo que incluimos elementos adicionales.

Luis: Para lo que fue el desarrollo de la etapa de exploración pude comprender mejor el manejo de los componentes léxicos que componen un archivo de texto, separándolo así por tokens para luego identificarlos dependiendo de la gramática que implementa nuestro lenguaje, además comprendí mejor el manejo de regex, diccionarios, y manejo de objetos en python.

Creación de componentes léxicos

Al hacer la prueba se va a imprimir en pantalla un cierto orden donde se va a reflejar el manejo de errores junto con la lectura de la gramática del Lenguaje Pandémico. De esta forma se puede reflejar visualmente cómo se manejan las pruebas.

Los componentes léxicos se imprimen en pantalla con el orden de siguiente:

1. Número de línea
2. Número de índice
3. Tipo de componente léxico
4. Texto del componente léxico
5. Atributos adicionales del componente

De esta forma se puede observar de forma muy concreta donde se produce el error.

Pruebas

El funcionamiento del programa trata de hacer pruebas a archivos .covid que estén con el formato del lenguaje Pandémico. Por lo que se tienen 6 archivos en los que se fue probando el funcionamiento del lenguaje y del manejo de errores. En esta parte va estar hablando del funcionamiento correcto y de errores. O sea, de manera de que las pruebas que se hicieron fueron con la intención de detectar errores, una vez habiendo programado las líneas de código en las que iba a entrar el manejo del error. Por los que se muestra a continuación unas pocas pruebas junto con el resultado, se marco con rojo donde hubo el error y se muestra lo que devuelve la prueba:

Archivo factorial.covid:

```
xxxfactorial
!virus Factorial num, n :{
    molecula &respuesta := 1.3
    cuarentena@ num #mismoke 1 :{
        respuesta @:= respuesta #por num
        num := num #quitar 1
    }empalidomuytemprano
    @vomito respuesta
}empalido
```

Resultado:

```
ERROR al procesar linea #2: !virus Factorial num, n :{ >>> Una variable esta contaminada con otros elementos en >>>
!virus
ERROR al procesar linea #3: molecula &respuesta := 1.3 >>> Una variable esta contaminada con otros elementos en >>>
&respuesta
ERROR al procesar linea #4: cuarentena@ num #mismoke 1 :{ >>> Una variable esta contaminada con otros elementos
en >>> @@
ERROR al procesar linea #5: respuesta @:= respuesta #por num >>> Una variable esta contaminada con otros
elementos en >>> @:=
ERROR al procesar linea #7: }empalidomuytemprano >>> El encapsulamiento esta enfermo en >>>
}empalidomuytemprano
```

#Linea	#Indice	Categoría	Componente	Descripción
2	1	ERRORVARIABLE	!virus	Una variable esta contaminada con otros elementos
3	4	ERRORVARIABLE	&respuesta	Una variable esta contaminada con otros elementos
4	3	ERRORVARIABLE	@	Una variable esta contaminada con otros elementos
5	4	ERRORVARIABLE	@:=	Una variable esta contaminada con otros elementos
6	2	IDENTIFICADOR	num	Identificadores o variables
6	4	ASIGNACION	:=	equivalente a =
6	6	IDENTIFICADOR	num	Identificadores o variables

```

6   8   OPERADOR      #quitar      Operador logico de resta -
6   10  MOLECULA       1           Esto es un numero entero
7   2   ERRORENCAPSULACION }empalidomuytemprano El encapsulamiento esta enfermo
8   2   FUNCION         @vomitito    Retorno todo, hasta el desayuno si queres
8   4   IDENTIFICADOR   respuesta     Identificadores o variables
9   1   ENCAPSULACION   }empalido    inicio con :{ y termino con }empalido

```

Archivo calcularimpuesto.covid:

```

virus CalcularImpuesto precioProducto, impuesto, 999descuento :{
    micro_molecula porcentaje := 0.12.21
    micro_molecula descuentoAux := (descuento #cortar 100) #por precioProducto)
    micro_molecula precioProductoAux := precioProducto #quitar descuentoAux
    micro_molecula impuestoAux := (impuesto #corrtar 100) #por precioProductoAux
    @vomitito impuestoAux
}empalido

```

Resultado:

ERROR al procesar linea #1: virus CalcularImpuesto precioProducto, impuesto, 999descuento :{ >>> Un parametro esta contaminada con otros elementos en >>> 999descuento

ERROR al procesar linea #2 micro_molecula porcentaje := 0.12.21 >>> Su flotante tiene varicela en >>> .21

ERROR al procesar linea #5: micro_molecula impuestoAux := (impuesto #corrtar 100) #por precioProductoAux >>> Una variable esta contaminada con otros elementos en >>> #corrtar

#Linea	#Indice	Categoría	Componente	Descripción
1	11	999descuento		Un parametro esta contaminada con otros elementos
2	9	ERRORFLOTANTE	.21	Su flotante tiene varicela
3	2	PALABRA_CLAVE	micro_molecula	Identificadores o variables
3	4	IDENTIFICADOR	descuentoAux	Identificadores o variables
3	6	ASIGNACION	:=	equivalente a =
3	8	PUNTUACION	(descuento	Esto es un signo de puntuacion
3	9	IDENTIFICADOR	descuento	Identificadores o variables
3	11	OPERADOR	#cortar	Operador logico de division /
3	13	MOLECULA	100)	Esto es un numero entero
3	14	PUNTUACION)	Esto es un signo de puntuacion
3	16	OPERADOR	#por	Operador logico de multiplicacion *
3	18	IDENTIFICADOR	precioProducto)	Identificadores o variables
3	19	PUNTUACION)	Esto es un signo de puntuacion
4	2	PALABRA_CLAVE	micro_molecula	Identificadores o variables
4	4	IDENTIFICADOR	precioProductoAux	Identificadores o variables
4	6	ASIGNACION	:=	equivalente a =
4	8	IDENTIFICADOR	precioProducto	Identificadores o variables
4	10	OPERADOR	#quitar	Operador logico de resta -
4	12	IDENTIFICADOR	descuentoAux	Identificadores o variables
5	11	ERRORVARIABLE	#corrtar	Una variable esta contaminada con otros elementos
6	2	FUNCION	@vomitito	Retorno todo, hasta el desayuno si queres
6	4	IDENTIFICADOR	impuestoAux	Identificadores o variables
7	1	ENCAPSULACION	}empalido	inicio con :{ y termino con }empalido

Manejo de errores

El manejo de errores estuvo compuesto por cuatro errores que se podrían encontrar en el Lenguaje Pandémico a lo programado en el explorador. Los cuales son el error de parámetro, flotante, variable y encapsulamiento. De esta forma se hicieron las pruebas para lograr el cometido con este entregable de poder manejar errores y a su vez poder informarle al usuario del error. La información que se decidió dar a conocer fue la línea donde paso el error, el tipo de componente en este caso qué tipo de error fue, además de la mostrar la palabra donde paso el error junto con el índice de la línea y una pequeña descripción del error. El manejo del error se refleja en la impresión donde se muestra el número de línea donde se produjo el error, el índice de esa línea donde se encontró el componente léxico que se dio el error, se muestra el tipo de componente, el texto del componente y una pequeña descripción.

Completitud del proyecto

El proyecto contenía como objetivo principal realizar un explorador con el fin de permitir manejar el lenguaje de pandémico y leer un documento con terminación .covid para analizar la escritura del usuario y validar si los elementos ingresados concuerdan con la sintaxis BNF desarrollada (además del manejo de errores?). Para la realización de este explorador fueron necesarias múltiples intervenciones y modificaciones al concepto original para adaptar el lenguaje a uno capaz de facilitarle la interacción y entendimiento al usuario y mediante el despliegue de mensaje de consola realizar un conjunto de elementos para mostrar los diferentes errores encontrados y las posibles soluciones de estos, basándonos siempre en la jerga del lenguaje de PANDÉMICO basados en el COVID-19 y conceptos médicos propuesto por el grupo en general.

Salud mental / Amor propio / Diversión / Crecimiento personal

En esta sección, cada uno de los integrantes del grupo adjunta una serie de memes lo cuales lo hacen sentir identificado con respecto al desarrollo de este proyecto. Además de poder adjuntar canciones.

Jeremy:

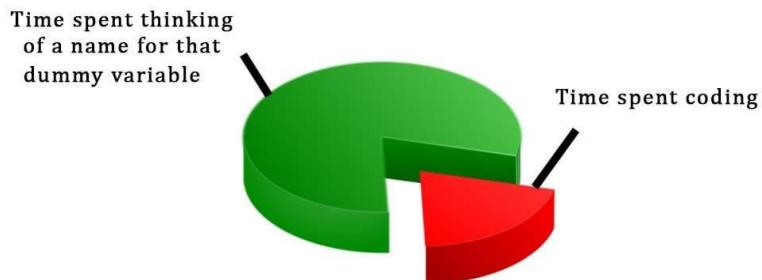
Compiler: Error at line 308

Me: "What? How? My code only has 40 lines

Compiler:



Programming Fact



Canciones.

https://open.spotify.com/track/7CbUZpSdZZ4EYKZ7Cdj7v?si=39ouJcO4Q3e-YudVnO7RqA&utm_source=copy-link

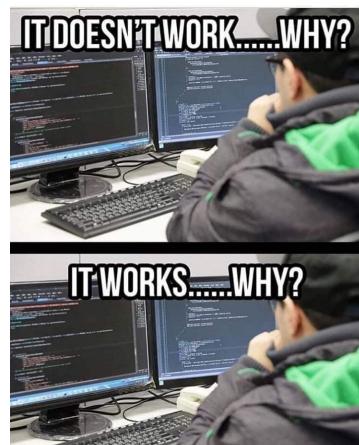
https://open.spotify.com/track/0vAnx8e7RK2sW1WycG2Ws6?si=YQONjU8ZQvSK3T-Oa2e-ag&utm_source=copy-link

https://open.spotify.com/track/0pa7rqyg7P2UfpA2Nu6dsI?si=LAU-ToNsQDeQBvnYyAnuzg&utm_source=copy-link

Alonso:



When your code compiles
after 253 failed attempts



Me after writing less than a
line of code:



Juan Pablo:



Luis Ortiz

